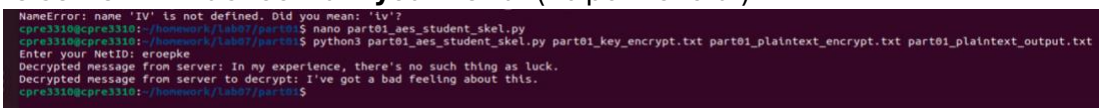
Lab 07 Template – Ethan Roepke

Part 01

1. **Take a screenshot of the response from the server and the decrypted message from the server. It must contain your netid.** (10 points total)



2. **What difficult problem about using symmetric keys does this demonstrate? Think about the two parties (your client and my server) having to share the same key. What protocol have we started to discuss in lecture that can solve this problem?** (20 points total, 10 points each question)

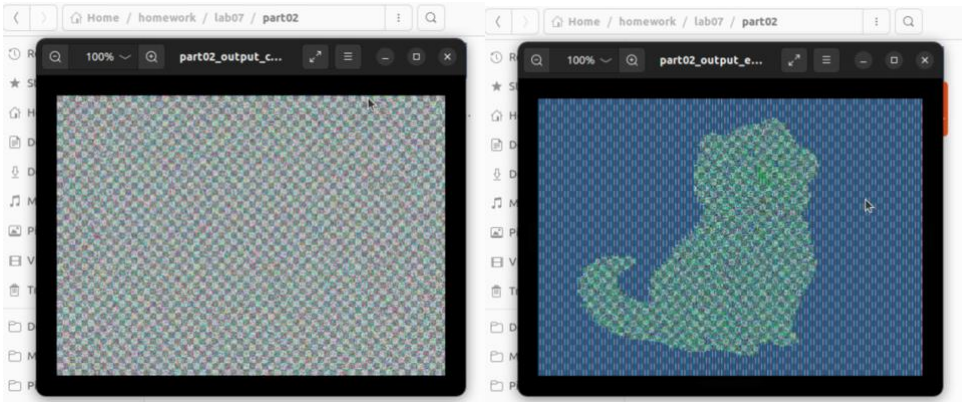
   The difficult problem about using symmetric keys between 2 parties, for instance me the client and the server, is sharing the key between each other, without the key being intercepted by a third party. A third party attacker getting the key will be able to decrypt all the messages. The protocol we started to discuss in class is Diffie Hellman Key Exchange Protocol. The protocol will allow both parties to generate the exact symmetric key instead of sharing the key.

3. **Functional code submitted to Canvas.** (20 points total)

   part01_eroepke_aes_skel.py

Part 02

4. **Open the two images and take a screenshot of them.** (10 points)



5. **How does ECB mode handle these repetitive blocks, and why does this lead to visible patterns in the encrypted image?** (10 points total, 5 points each question)

   ECB mode handles repetitive blocks by encrypting each block of the plaintext independently and if plaintext blocks encounter identical pieces, then the cipher text will encounter identical pieces. This leads to visible patters when we worked with the images because ECB does not use any chaining in encryption. So identical pieces in plaintext will have the same identical pieces in the cipher text.

6. **What effect does CBC mode have on repetitive structures in the image compared to ECB mode? How does the introduction of the Initialization Vector (IV) in CBC mode help in obscuring patterns that are visible in ECB mode?** (20 points total, 10 points each question)

   The effects that CBC have on repetitive structure in the image compared to ECB mode is that CBC uses chaining and the plaintext piece being encrypted is dependent on the previous piece. So being dependent on the previous piece of encryption means no part of the cipher text will be identical because of the influence of previous blocks. Compared to ECB which is completely independent so you will see more repetitive pieces in plaintext and ciphertext. The introduction of IV in CBC mode help obscure the patterns that were visible in ECB mode because IV adds a random value and XORs the first encryption from the plaintext. The IV is always different for every encryption so if you encrypt the same plaintext over and over, you will receive different ciphertext outputs.

7. **Upload python code for part 02**.(10 points)

   part02_eroepke_ecb_cbc.py